IN THE SPECIFICATION

Please replace the original paragraphs of the same number with the following replacement paragraphs:

--[0003] The widespread digitization of media (including multimedia) content, especially by the consumer segment, coupled with the growth in digital communication networks and easier methods to transfer digital content is changing the nature of media content delivery and usage. Media content can now be captured and encoded in one or more of a plurality of digital formats (*e.g.*, MPEG, Windows Media Format, VCD, *etc.*) distributed over digital networks such as the ~~internet~~ Internet or on digital media and accessed using general purpose computing equipment or special purpose equipment.--

--[0010] Fig. 1 is an illustration of an environment in an exemplary implementation in which a computer provides access to a plurality of media[[.]];--

--[0011] Fig. 2 is a high level block diagram of a system in an exemplary implementation in which the system, implemented in software, includes an application that interacts with a media foundation to control presentation of a plurality of media[[.]];--

--[0012] ~~Fig. 3 is~~ Figs. 3A-3B are a schematic illustrations ~~illustration~~ of an exemplary partial topology;--

--[0028] The media foundation 204 [[202]] may utilize several components among which include the media timeline 122, the timeline source 124, a media source 210, a media processor 212, a media session 214, the media engine 208, a source resolver 216, one or more transforms 218, one or more media sinks 220, 222, and so on. One advantage of various illustrated and described embodiments is that the system 200 is a pluggable model in the sense that a variety of different kinds of components can be utilized in connection with the systems described herein. Also included as a part of system 200 is a destination 224, which is discussed in more detail below. In at least one embodiment, however, the destination 224 is an object that defines where a presentation is to be presented (e.g. a window, disk file, and the like) and what happens to the presentation. That is, the destination may correspond to one or more of the media sinks 220, 222 into which data flows.--

--[0029] The media timeline 122 employs a timeline object model which provides a way for a user to define a presentation based on media that is rendered by the timeline source 124. The media timeline 122 may range from a sequential list of media files to more complex forms. For example, the media timeline 122 may employ file structures, such as SMIL (Synchronized Multimedia Integration Language) and AAF (Advanced Authoring Format), to express media playback experiences that include transitions between media, effects, and so on. The application 202, for instance, may be configured as a media player that can play a list of songs, which is commonly referred to as a playlist. As another example, in an editing system a user may overlay one video over the other, clip a

media, add effect to the media and so forth. Such groupings or combinations of media may be expressed using the media timeline 122.--

--[0040] The timeline plugin 234 may also provide an interface such that the application 202 may interact with the timeline plugin directly, such as to load and save the media timeline 122 from or to a file. For example, the timeline plugin 234 may be created and then called to initiate a load function to provide a bytestream. The timeline plugin 234 may then parse the file and create a root node and any additional nodes to create the media timeline 122. The timeline plugin 234 may also be used to persist the media timeline 122 to different formats. For example, the application 202 may create the media timeline 122 programmatically. In other words, the application may act as the timeline generator 120 of Fig. 1. The application 202 may then create a timeline plugin for ASX (Advanced Stream Redirector) files, and ask the timeline plugin to save the media timeline 122 in the ASX format. In another example, a user can open an m3u file, i.e. a playlist file format for specifying multiple MP3 files, get the media timeline 122 from it, and then ask the timeline plugin to save the media timeline 122 in the ASX format. In this way, the media foundation 204 may expose a plurality of software components that provide media functionality over an application programming interface for use by the application 202.—

--[0076] This method permits an application 202 to specify a "smart" connector callback to the topology loader 232. This "smart" connector gives the application the flexibility to influence the topology loader 232 during the process of constructing a topology. By way

of example, the application may be given the chance to connect two nodes before the topology loader 232 attempts to connect the nodes. Alternatively the application may have the ability chance to create a certain DMO (DirectX Media Objects) before the topology loader 232 attempts to create the DMO. --

--[00157] At operation 515 the topology loader determines whether there are corresponding nodes in the partial topology and the previous full topology. In an exemplary implementation this may be performed by comparing the nodes in the partial topology and the previous full topology received as parameters. If there are no corresponding nodes, then control passes to operation 520 and object caching operations 500 terminate. The new topology may then be instantiate using conventional techniques. Exemplary techniques are disclosed in corresponding U.S. Patent Application Serial No. ~~XX/XXX,XXX~~ 10/796,505, entitled RESOLVING PARTIAL MEDIA TOPOLOGIES, commonly assigned to Microsoft Corporation of Redmond, Washington, USA, the disclosure of which is incorporated herein by reference.--